Defense Information Infrastructure (DII)

Common Operating Environment (COE)

Statement of Functionality (SOF)
for the
Latitude-Longitude-Time (LLT) Observation
Application Program Interface (API) Segment (MALLT)
Release 4.3 Series

8 July 1999

Prepared for:
Space and Naval Warfare Systems Command
Environmental Systems Program Office
(SPAWAR PMW-185)

Prepared by:
Fleet Numerical Meteorology and Oceanography Center
Monterey, CA

and

Integrated Performance Decisions, Inc.
Monterey, CA

Table of Contents

List of Tables

List of Figures

1 SCOPE

1.1 Identification

This document describes the functionality of the Latitude–Longitude–Time (LLT) Observation Application Program Interface (API) (MALLT) Segment, Version 4.3 Series, of the Tactical Environmental Data Server (TEDS) component of the Navy Integrated Tactical Environmental Subsystem (NITES). The MALLT segment provides APIs for the storage and retrieval of METOC point observations.

1.2 System Overview

TEDS is the Meteorological and Oceanographic (METOC) Database for the NITES system. It is a DII COE shared database, with separate segments for different types of data and different functionalities. For each data type, there is a Database segment that provides the database schema and "static" database tables such as the Master Station Library in the LLT Observation database. There is also an Application Program Interface (API) segment that provides programming interfaces to the database. The API segments provide functionality for programs to cause data to be ingested into the database, to get a catalog of data in the database, to retrieve data by query or by individual item ID, to delete data from the database, etc. Underlying all of TEDS is a Commercial Off-the-Shelf (COTS) Relational Database Management System (RDBMS). This is currently Informix v7.2x, but could be any RDBMS.

1.3 Document Overview

Section 2 provides a more detailed overview of TEDS, while Section 3 contains more detail concerning the specific functionality provided by the MDLLT segment.

2 TEDS FUNCTIONALITY OVERVIEW

The software described in this document forms a portion of the TEDS component of NITES. On 29 October 1996, the Oceanographer of the Navy issued a Program Policy statement in letter 3140 Serial 961/6U570953, modifying the Program by calling for five seamless software versions that are DII COE compliant, preferably to level 5.

The five versions are:

- NITES Version I — The local data fusion center and principal METOC analysis and forecast system
- NITES Version II — The subsystem on the Joint Maritime Command Information System (JMCIS) or Global Command and Control System (GCCS) (NITES/Joint METOC Segment (JMS))
- NITES Version III — The unclassified aviation forecast, briefing, and display subsystem tailored to Naval METOC shore activities (currently satisfied by the Meteorological Integrated Data Display System (MIDDS))
- NITES Version IV — The Portable subsystem composed of independent Personal Computers (PCs)/workstations and modules for forecaster, satellite, communications, and Integrated Command, Control, Communications, Computer, and Intelligence Surveillance Reconnaissance (IC4ISR) functions (currently the Interim Mobile Oceanographic Support System (IMOSS))
- NITES Version V — Foreign Military Sales (currently satisfied by the Allied Environmental Support System (AESS))

NITES I acquires and assimilates various METOC data for use by US Navy and Marine Corps weather forecasters and tactical planners. NITES I provides these users with METOC data, products, and applications necessary to support the warfighter in tactical operations and decision making. NITES I provides METOC data and products to NITES I and II applications, as well as other systems requiring METOC data, in a heterogeneous, networked computing environment.

The NITES I Concept of Operations and system architecture require that the METOC Database be distributed both in terms of application access to METOC data and products and in terms of physical location of the data repositories. The organizational structure of the database is influenced by these requirements, and the components of this distributed database are described below.

In accordance with DII COE database concepts, the METOC Database is composed of six DII COE-compliant *shared database* segments. Associated with each shared database segment is an Application Program Interface (API) segment. The segments are arranged by data type as follows:

Data Type	**Data Segment**	**API Segment**
Grid Fields	MDGRID	MAGRID
LLT Observations	MDLLT	MALLT
Textual Observations and Bulletins	MDTXT	MATXT
Remotely Sensed Data	MDREM	MAREM
Imagery	MDIMG	MAIMG
Climatology Data	Segments named by data type. To date, only DBDB-V segments (MDDBV and MADBV) have been released.	

A typical client-server installation is depicted in Figure 2-1 on the next page. This shows the shared database segments residing on a DII COE SHADE database server, with a NITES I or II client machine hosting the API segments. Communication between API segments and shared database segments is accomplished over the network using American National Standards Institute (ANSI)-standard Structured Query Language (SQL).

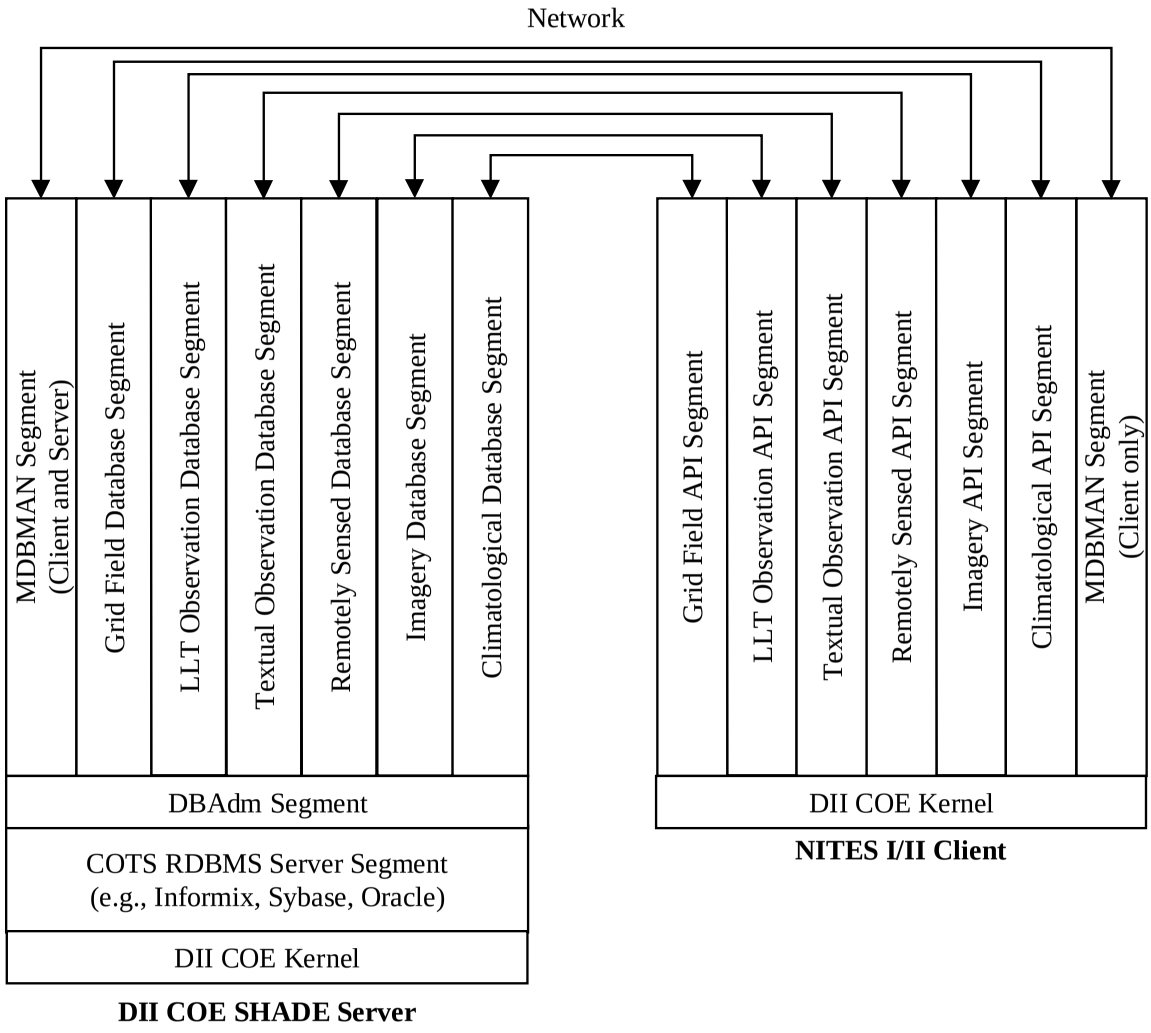

Figure 2-1. NITES METOC Database Conceptual Organization

3 MALLT SEGMENT FUNCTIONALITY

The APIs in the MALLT segment deal with point observations. These include surface weather observations (hourlies, specials, synoptic observations, METAR reports Terminal Aerodrome Forecasts (TAFs), etc.), upper air observations (e.g., radiosonde reports, aircraft observations), and ocean soundings (bathythermograph, sound velocity profiles, etc.). For upper air and ocean soundings, the database may also store data derived form the original soundings in the form of upper air profiles and ocean profiles.

The methods are associated with the database in general and not each class. The purpose of this is to use the relational functionality provided by the Relational Database Management System (RDBMS) to implement the data model.

1. **MALLTIngest**: Adds an object to the database.

2. **MALLTDeleteByID**: Deletes objects from the database based on given criteria.

3. **MALLTCatalog**: Provides summary information about observations in the database.

4. **MALLTGetByID**: Retrieves a single object returned as a result of a catalog.

5. **MALLTGetByQuery**: Retrieves multiple objects.

6. **MALLTUpdateByID**: Updates limited attributes within an object.

7. **MALLTDeleteByQuery**: Allows purging from the database of observations matching query criteria.

8. **MALLTFreeLL**: Used to free a linked list returned by a query operation.

9. **MALLTGetStationByArea**: Receives a geographic area, queries the database for stations residing there, then returns a list of the stations found.

10. **MALLTGetStationByID**: Receives a station ID and returns detailed station information.

Each of the methods, with the exception of the Ingest methods, allows the input of selection criteria. The criteria are used to construct SQL statements to retrieve or otherwise manipulate objects that match the criteria. The Ingest methods do not provide any criteria because they are simply writing data to the database.

Five other methods are also provided to manage database connections:

1. **MALLTConnect**: Connects to the default LLT database.
2. **MALLTRemoteConnect**: Connects to the specified LLT database.
3. **MALLTDisconnect**: Disconnects from the default LLT database.
4. **MALLTRemoteDisconnect**: Disconnects from the specified LLT database.
5. **MALLTSetConnection**: Sets the current connection to the specified LLT database.

Table 3-1 provides a list of all APIs for LLT observation data.

Table 3-1. LLT Observation Data APIs

API Name	Functional Description
MALLTCatalog	Catalogs observations based on selection criteria.
MALLTConnect	Connects to the default LLT Observation Database.
MALLTDeleteByID	Deletes an observation from the database.
MALLTDisconnect	Disconnects from the LLT Observation Database.
MALLTGetByID	Gets a single observation based on information returned from a catalog query.
MALLTGetByQuery	Retrieves groups of observations based on selection criteria.

Table 3-1. LLT Observation Data APIs

API Name	Functional Description
MALLTGetStationByArea, MALLTGetStationByID	Gets station IDs and geographic area by querying the MDLLT database.
MALLTIngest	Adds an observation to the database.
MALLTFreeLL	Frees a linked list returned by a query operation.
MALLTDeleteByQuery	Purges observations from the database based on selection criteria.
MALLTRemoteConnect	Connects to the specified LLT Observation Database.
MALLTRemoteDisconnect	Disconnects to the specified LLT Observation Database.
MALLTSetConnection	Sets the active connection to the specified LLT Observation Database.
MALLTUpdateByID	Updates an observation based on given criteria.